# DORMITORY MANAGEMENT SYSTEM

**By**

Ashfaqur Rahman Tokee (Exam Roll: 192339)
Md. Shakil Hossain (Exam Roll: 192340)
Mahabubur Rahman (Exam Roll: 192341)
Nahidul Islam (Exam Roll: 192345)

A project report submitted to the Institute of Information Technology
in partial fulfilment of the requirements for the degree of
Bachelor of Science in Information Technology

**Supervised By**

**Fahima Tabassum**

Professor

Institute of Information Technology

Jahangirnagar University


Institute of Information Technology
Jahangirnagar University
Savar, Dhaka-1342
29 March, 2023

# DECLARATION

This industrial tour report is submitted to the Institute of Information Technology, Jahangirnagar University, Savar, Dhaka in partial fulfillment of the requirements for having the B.Sc. (Hons.) degree in ICT. This is also needed to certify that the project work is under the 3rd Year 2nd Semester course of the IIT "ICT-3200: Project Work and Course Viva". So, we are here declaring that this project report has not been submitted elsewhere for the requirement of any kind of degree, diploma or publication.

**Ashfaqur Rahman Tokee**
**2022**

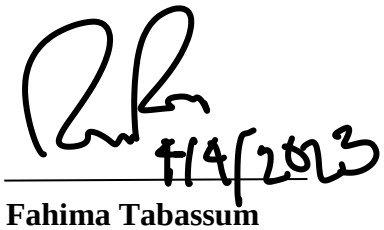
**Md. Shakil Hossain**
**2023**

**Mahabubur Rahman**
**2024**

**Nahidul Islam**
**2028**

# CERTIFICATE

The project titled “Dormitory Management System” submitted by Ashfaqur Rahman Tokee (Exam Roll: 192339),Md. Shakil Hossain (Exam Roll: 192340), Mahabubur Rahman (Exam Roll: 192341), Nahidul Islam (Exam Roll: 192345), Session: 2018-2019, has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Bachelor of Science in Information Technology on 30 March, 2023.



**Fahima Tabassum**
Professor,
Institute of Information Technology,
Jahangirnagar University,
Savar, Dhaka – 1342.

Accepted and approved in partial fulfilment of the requirement for the degree Bachelor of Science (honors) in Information Technology

## BOARD OF EXAMINERS

_____

Dr. Jesmin Akhter — Chairman
Professor,
Institute of Information Technology,
Jahangirnagar University,
Savar, Dhaka – 1342.

Member

_____

Dr. Md. Sazzadur Rahman,
Associate Professor,
Institute of Information Technology,
Jahangirnagar University,
Savar, Dhaka – 1342.

_____

Dr. Rashed Mazumder, — Member
Assistant Professor,
Institute of Information Technology,
Jahangirnagar University,
Savar, Dhaka – 1342.

_____

Dr. Kazi Muheymin-Us-Sakib, — Member (External)
Professor,
Institute of Information Technology,
University of Dhaka.

# DEDICATION

We dedicate this Dormitory Management System project to all the students who strive for a comfortable and efficient living experience in their college dormitories. Our goal was to develop a system that streamlines the management of dormitories, ensuring that students have easy access to essential services and facilities.

We also dedicate this project to the dormitory managers and administrators who work tirelessly to meet the needs of their residents. We hope that this system will help them to manage their duties more effectively, allowing them to provide a better quality of life for their students.

Finally, we would like to dedicate this project to our families and loved ones who have supported us throughout this endeavor. Their encouragement and belief in us have been instrumental in our success.

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGEMENT

First of all we would like to thank the Almighty for giving us the opportunity to complete this work successfully. Our acknowledgement is meant to express our sincere gratitude to all those people who have been associated with this project and have helped us with it and by sharing their experiences and valuable opinions through which we received the required information crucial for our project. We are thankful to our parents for their relentless support. Most importantly we are grateful to our honorable supervisor who took time out to guide us and provide us with all the necessary materials and sufficient knowledge that was the major requirement.

Finally, we convey our regards to our honorable teacher **Professor Fahima Tabassum** Mam for giving us the opportunity to learn the subject particularly practically.

# ABSTRACT

This report describes the development of a dormitory management system website designed to improve the efficiency and effectiveness of managing dormitories or student housing. The website addresses the specific challenges faced by dormitory management, such as managing room assignments, maintaining accurate occupancy records, and processing billing and payment information. The proposed system provides user registration and login, room assignment management, and payment processing functionality. The website was implemented using a combination of programming languages and frameworks, and rigorous testing and validation processes were performed to ensure its reliability and effectiveness. The results of the project showed that the website provided significant benefits to both students and administrators, such as reducing paperwork, increasing transparency, and improving communication. The report concludes with potential areas for future development and improvement, such as integrating with other campus management systems or adding new features to the website. Overall, the dormitory management system website has the potential to revolutionize the management of dormitories and student housing, and provide significant benefits to both students and administrators.

# TABLE OF CONTENTS

# CHAPTER 1
# INTRODUCTION

## 1.1 Introduction

A Dormitory Management System (DMS) is a software application designed to manage and automate the operations of dormitories or hostels. It's a centralized system that simplifies the management tasks of student boarding facilities, making it easier for administrators to manage students, rooms, inventory, and other aspects of running a dormitory.

The DMS typically consists of several modules that cater to different aspects of dormitory management, such as admissions, accommodation, meal plans, payments, maintenance, and security. The system may also include features such as online booking, room assignment, student profiles, attendance tracking, and communication tools, among others.

The primary objective of the Dormitory Management System is to simplify the management of a dormitory and streamline its operations. By automating routine tasks, reducing paperwork, and providing real-time information, the system helps administrators to save time, reduce errors, and enhance the overall experience of students living in the dormitory.

Dormitory Management Systems are used by educational institutions, such as universities, colleges, boarding schools, and hostels, as well as private organizations that provide student housing. With the growing demand for student housing and the increasing complexity of managing dormitories, DMS has become an essential tool for effective dormitory management.

## 1.2 Objectives

The objectives of the "Dormitory Management System" are:

- To provide a digital management system for JU hall.
- To develop a user-friendly and intuitive website for dormitory management that can be accessed and utilized by both students and administrators.
- To improve the efficiency of managing dormitories by providing automated tools for managing room assignments, occupancy records, and billing and payment processing.
- To reduce paperwork and administrative burden by providing a digital platform for managing dormitories.
- To provide accurate and up-to-date information on dormitory occupancy, room assignments, and billing and payment information.
- To ensure the security and confidentiality of student and administrator data by implementing robust security measures.
- To provide a reliable and scalable system that can accommodate future growth and expansion.
- To improve the overall student experience by providing a streamlined and efficient process for managing dormitories.

## 1.3 Expected Outcome

The expected outcomes of the "Dormitory Management System" are:

- A complete digital management system for JU Hall.
- Digital means of storing student information.
- Digitalized and Secure communication with authority and student.
- Reduced paperwork and administrative burden through the use of a digital platform for managing dormitories.
- Improved accuracy and timeliness of information related to dormitory occupancy, room assignments, and billing and payment information.
- Improved student experience through a streamlined and efficient process for managing dormitories.
- Potential cost savings through reduction of administrative costs and improved management of dormitories.
- Reduction of human resource and additional cost of JU Hall.

# CHAPTER 2 BACKGROUND

## 2.1 Introduction

Managing dormitories or student housing can be a challenging and complex task. Dormitories have a significant impact on the academic and social experience of students, and are therefore an important aspect of campus life. However, managing dormitories involves a range of administrative tasks, such as managing room assignments, maintaining accurate occupancy records, and processing billing and payment information, that can be time-consuming and difficult to manage efficiently. Furthermore, traditional manual methods of managing dormitories can be cumbersome and error-prone, leading to potential problems such as inaccurate records, lost paperwork, and administrative bottlenecks.

To address these challenges, many universities and colleges are turning to digital solutions such as dormitory management system websites to streamline and automate dormitory management processes. These websites provide a centralized platform for managing dormitories, and can improve the efficiency, accuracy, and transparency of dormitory management. Such websites are typically designed to provide automated tools for managing room assignments, occupancy records, and billing and payment processing. In addition, they may provide features such as online student registration, communication channels between students and administrators, and other functionality that helps to streamline the dormitory management process.

Given the potential benefits of dormitory management system websites, many universities and colleges are exploring ways to implement these systems within their own campuses. The development of a dormitory management system website for a university can improve the overall management of dormitories, reduce administrative costs, and improve the overall student experience. In this project, we aim to develop a dormitory management system website for our university that addresses the specific challenges faced by our dormitory management team and provides an effective and efficient platform for managing dormitories.

## 2.2 Related Work

### 2.2.1 Dhaka University Hall

This website provides information about the university's Halls of Residence, including Salimullah Muslim Hall, Dr. Muhammad Shahidullah Hall, Jagannath Hall, and others. The website also provides information about the university's history, leadership, governance framework, and useful links for students and staff. Additionally, there are links to login pages for students and staff, as well as links to the university's tender notices and job postings.

Website Link: [Home :: Dhaka University (du.ac.bd)]

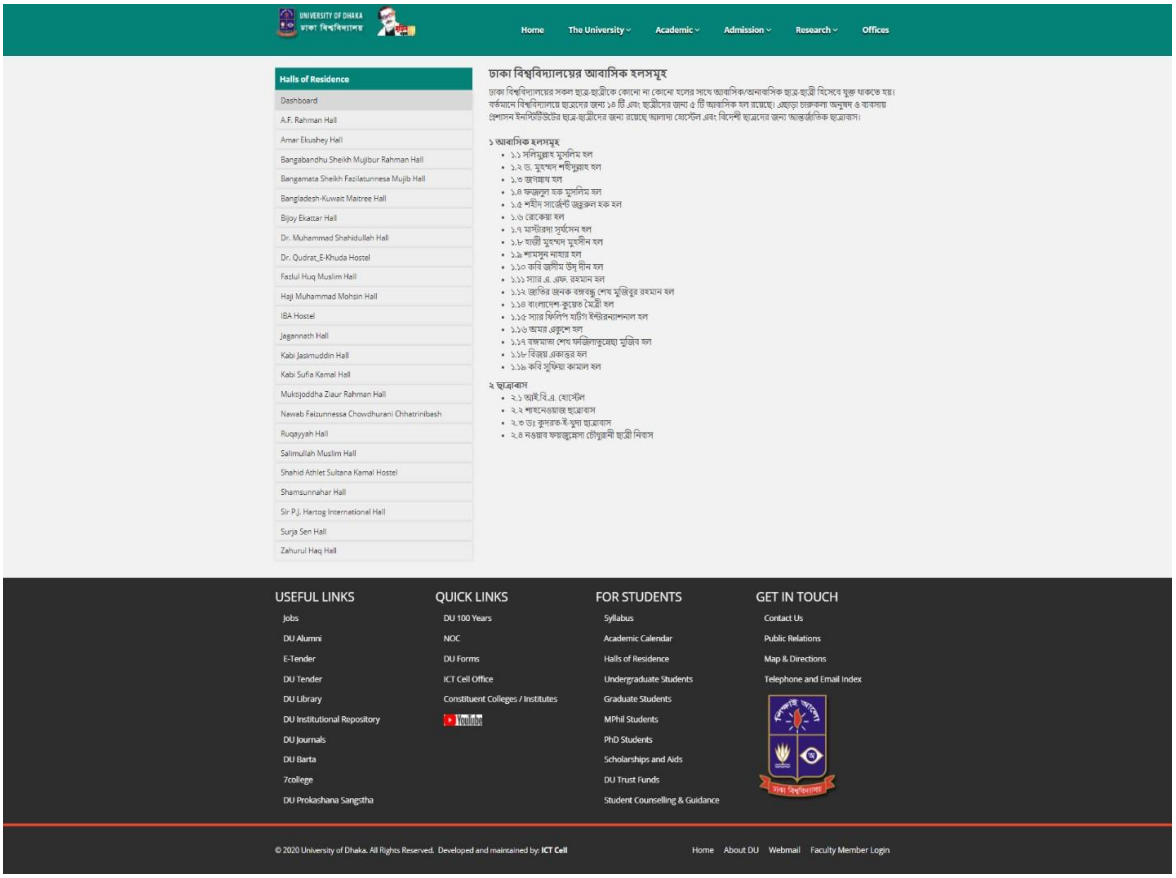



Figure 2.2.1 Home Page of Dhaka University

### 2.2.2 Shahjalal University of Science & Technology Hall

A simple website is used by Shahjalal University of Science & Technology Hall.

Website Link : [Shahjalal University of Science & Technology (sust.edu)]

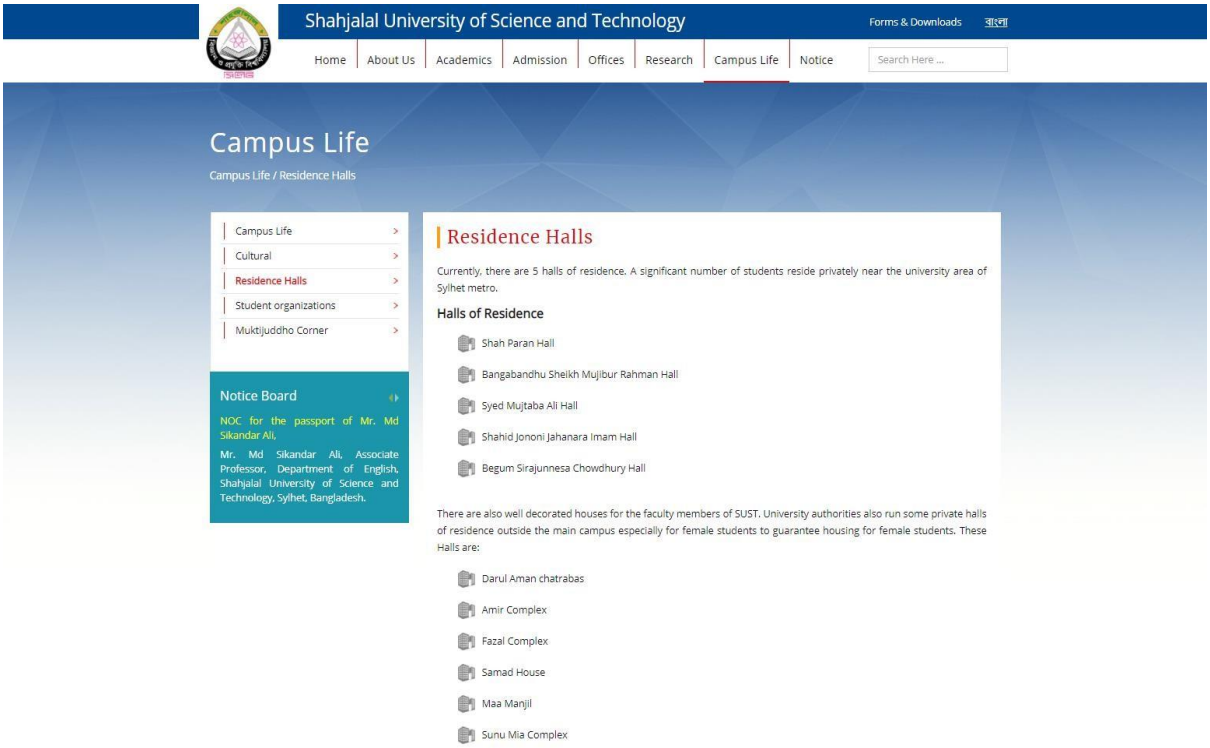



Figure 2.2.2 Home Page of Shahjalal University of Science and Technology

### 2.2.3 Khulna University of Engineering Technology

Digital world being digital by using information technology like many web-based automation system. A Web-based Dormitory Management System is designed and implemented by Khulna University of Information and Technology.


Figure 2.2.3 Home Page of Khulna University of Engineering and Technology

# CHAPTER 3

# SOFTWARE REQUIREMENTS SPECIFICATION

## 3.1 Requirement Collection and Analysis

### 3.1.1 Functional System Requirement:

This section gives a functional requirement that are applicable for “Dormitory Management System for JU Hall”.

- User registration and login: The website should have a registration system for both students and administrators to create user accounts, and a login system to authenticate users.
- Room management: The website should provide automated tools for managing room assignments, including the ability to assign rooms based on student preferences, gender, and other criteria.
- Occupancy records management: The website should provide a centralized platform for managing occupancy records, including the ability to view current occupancy, upcoming vacancies, and room change requests.
- Billing and payment processing: The website should provide automated billing and payment processing functionality, including the ability to view and pay bills online, manage payment plans, and track payment history.
- Communication channels: The website should provide communication channels between students and administrators, including messaging functionality, announcements, and alerts.
- Reporting and analytics: The website should provide reporting and analytics functionality to allow administrators to track occupancy rates, room utilization, and other key performance indicators.

These are some modules present in the website.

- Administrator module
- User Module
- Hostile Module
- Registration Module

### 3.1.2 Non-Functional System Requirements:

- Usability: The website should be easy to use and navigate for both students and administrators, with an intuitive user interface and clear instructions.
- Performance: The website should be fast and responsive, with quick loading times and minimal lag, even during periods of peak usage.
- Reliability: The website should be reliable and available 24/7, with minimal downtime or service disruptions.
- Compatibility: The website should be compatible with a wide range of devices and browsers, including desktops, laptops, tablets, and smartphones.
- Security: The website should be secure, with robust authentication and authorization mechanisms to protect user data and prevent unauthorized access.
- Privacy: The website should respect user privacy, with clear privacy policies and procedures in place to protect user data.
- Scalability: The website should be scalable, able to handle increasing amounts of data and traffic as the number of students and dormitories grows over time.
- Maintainability: The website should be maintainable, with clear and well-documented code that is easy to update and modify as needed.
- Accessibility: The website should be accessible to all users, including those with disabilities, by adhering to web accessibility guidelines and standards.
- Performance monitoring and reporting: The website should have monitoring and reporting mechanisms in place to track performance, identify issues, and resolve them quickly.

## 3.2 Use Case Modeling

### 3.2.1 Use Case Diagram

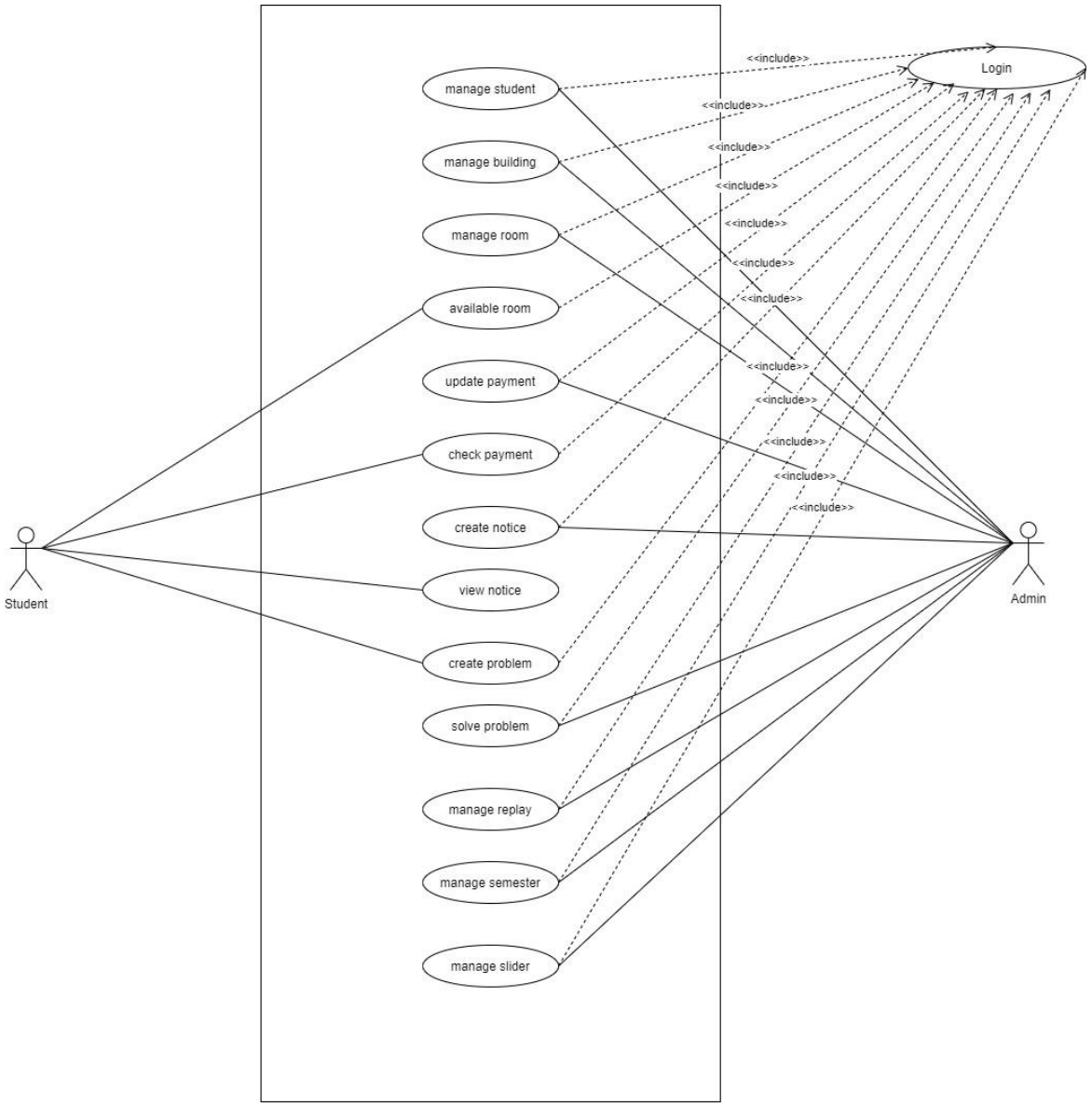



Figure 3.2.1 Use Case Diagram

## 3.3 Use Case Description

TABLE 3.3.1 Student Manage

| | |
|---|---|
| **Use Case Term** | Student Manage |
| **Actors** | Admin |
| **Flow of Events** | 1. Affix to Student 2. Remove Student 3.View Student Details |
| **Substitute Flows** | No student found<br>2. Do not add new student3.Invalid Information |
| **Pre-Condition** | Login |
| **Post Condition** | Confirm Student, Delete Student |

TABLE 3.3.2 Building Manage

| | |
|---|---|
| **Use Case Term** | Building Manage |
| **Actors** | Admin |
| **Flow of Events** | 1. Affix to Building 2. Upgrade Building 3. Remove Building |
| **Substitute Flows** | Chosen the wrong building<br>Building not found |
| **Pre-Condition** | Login |
| **Post Condition** | Chosen right building |

TABLE 3.3.3 Room Manage

| | |
|---|---|
| **Use Case Term** | Room Manage |
| **Actors** | Admin |
| **Flow of Events** | 1. Affix on Room 2. Upgrade Room 3.Remove Room |
| **Substitute Flows** | 1.Chosen false room 2.Delete incorrect room<br>3.Invalid Input |
| **Pre-Condition** | Login |
| **Post Condition** | Select the right room |

TABLE 3.3.4 Available Room

| | |
|---|---|
| **Use Case Term** | Available Room |
| **Actors** | Admin, Student |
| **Flow of Events** | 1.Check Available Room 2.Upgrade Room |
| **Substitute Flows** | Chosen false room<br>2. Don't updated available room3.Invalid Input's |
| **Pre-Condition** | Login |
| **Post Condition** | Select Building, Select Room |

TABLE 3.3.5 Update Payment

| **Use Case Term** | Update Payment |
|---|---|
| **Actors** | Admin |
| **Flow of Events** | 1.Affix to Payment 2.Upgrade Payment 3.Remove Payment |
| **Substitute Flows** | 1.Update wrong student payment 2.Don't delete running student payment<br>3.Invalid Input's |
| **Pre-Condition** | Login |
| **Post Condition** | Select correct student |

TABLE 3.3.6 Check Payment

| **Use Case Term** | Check Payment |
|---|---|
| **Actors** | Student |
| **Flow of Events** | 1. Check current payment 2. Check payment list |
| **Substitute Flows** | Incorrect student id<br>Invalid Input's |
| **Pre-Condition** | Login |
| **Post Condition** | Enter Login Information |

TABLE 3.3.7 Create Notice

| **Use Case Term** | Create Notice |
|---|---|
| **Actors** | Admin |
| **Flow of Events** | 1.Affix to Notice 2.Remove Notice |
| **Substitute Flows** | Affix wrong notice<br>Invalid Information |
| **Pre-Condition** | Login |
| **Post Condition** | Enter notice title and description |

TABLE 3.3.8 View Notice

| **Use Case Term** | View Notice |
|---|---|
| **Actors** | Student |
| **Flow of Events** | 1.View current Notice 2.View previous Notice |
| **Substitute Flows** | Select invalid notice<br>Can’t view future notice |
| **Pre-Condition** | Login |
| **Post Condition** | View notice panel |

TABLE 3.3.9 Create Problem

| **Use Case Term** | Create Problem |
|---|---|
| **Actors** | Student |
| **Flow of Events** | 1.Affix to Problem 2.Upgrade Problem 3.Remove Problem |
| **Substitute Flow** | Incorrect student id<br>Invalid Information |
| **Pre-Condition** | Login |
| **Post Condition** | View problem list, view reply of admin |

TABLE 3.3.10 Solve Problem

| **Use Case Term** | Solve Problem |
|---|---|
| **Actors** | Admin |
| **Flow of Events** | 1.Reply |
| **Substitute of Flow** | Can’t ignore problem<br>Invalid reply |
| **Pre-Condition** | Login |
| **Post Condition** | View problem solving list |

TABLE 3.3.11 Apply Manage

| **Use Case Term** | Apply Manage |
|---|---|
| **Actors** | Admin |
| **Flow of Events** | 1.Confirm Apply |
| **Substitute Flows** | Don’t update previous apply information<br>Invalid Information Input |
| **Pre-Condition** | Login |
| **Post-Condition** | View Apply list |

TABLE 3.3.12 Semester Manage

| **Use Case Term** | Semester Manage |
|---|---|
| **Actors** | Admin |
| **Flow of Events** | 1.Affix to Semester 2.Remove Semester 3.Upgrade Semester |
| **Substitute of Flows** | 1.Affix false semester 2.search wrong semester<br>3.Invalid Information |
| **Pre-Condition** | Login, Create a new semester |
| **Post Condition** | View All Semester |

## 3.4 Database Schema

Admin (A_ID, Password)

Provost (P_ID, Name, Email, Phone)

Staff (St_ID, Name, Phone, Email, Post, Salary, Address)

Student (S_ID, Name, Phone, Email, Registration, Batch, Department)

Room (R_ID, Block, Floor, No of Bed, S_ID)

Dinning (D_ID, Meal Type, Meal Rate, Menu, S_ID, St_ID)

Hall Charge (H_ID, Charge, A_ID, S_ID)

Facility (F_ID, PR_ID, ER_ID, RR_ID, S_ID)

Prayer Room (S_ID, PR_ID)

Entertainment Room (ER_ID, S_ID, TV, Table Tennis)

Reading Room (RR_ID, S_ID)

## 3.5 Diagram of Entity Relationship

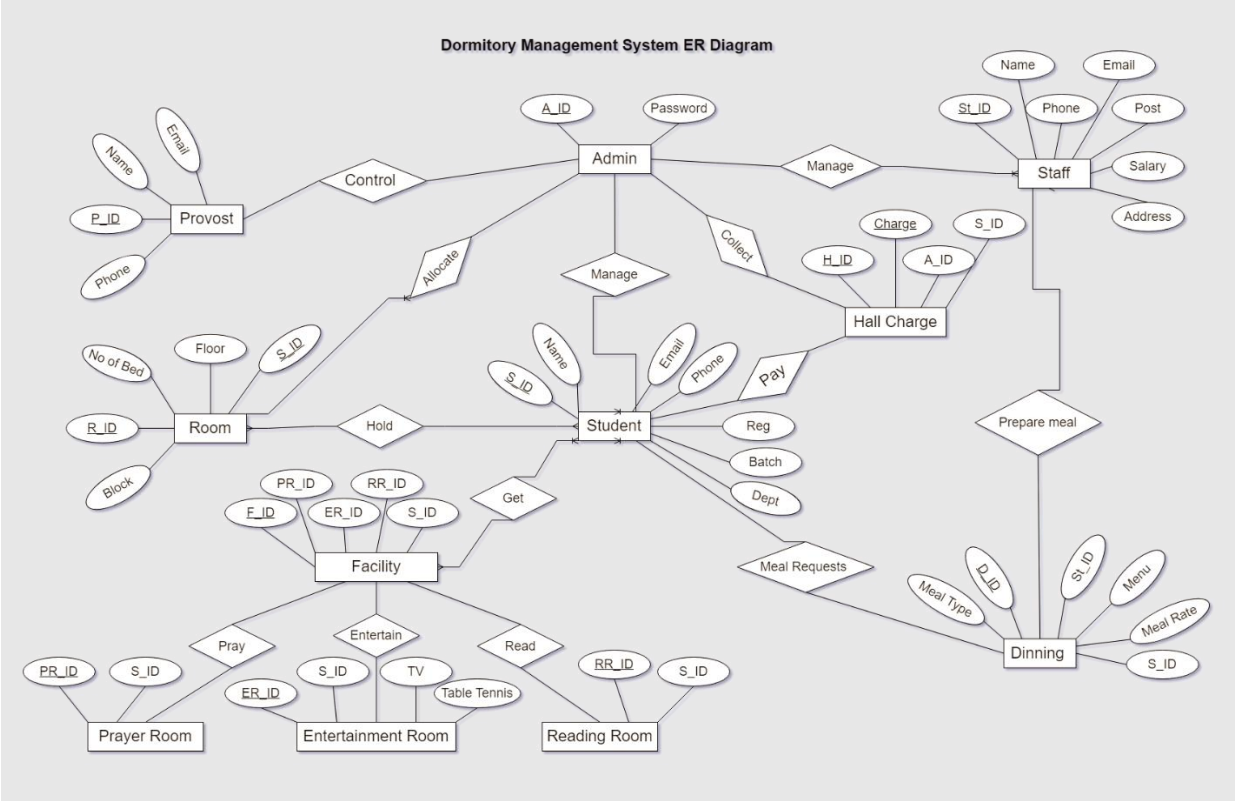



Figure 3.5 Entity Relationship

## 3.6 Diagram of DFD

### 3.6.1 Context Level


Figure 3.6.1 DFD Context Level

### 3.6.2 0 Level Diagram

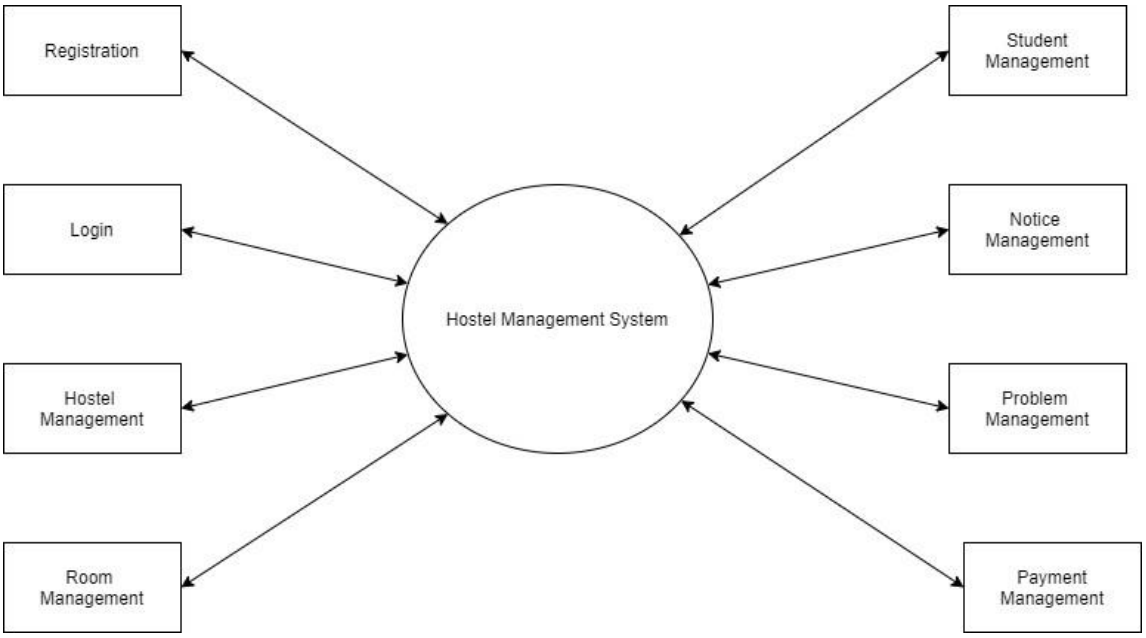



Figure 3.6.2 DFD 0 Level

### 3.6.3 1 Level Diagram

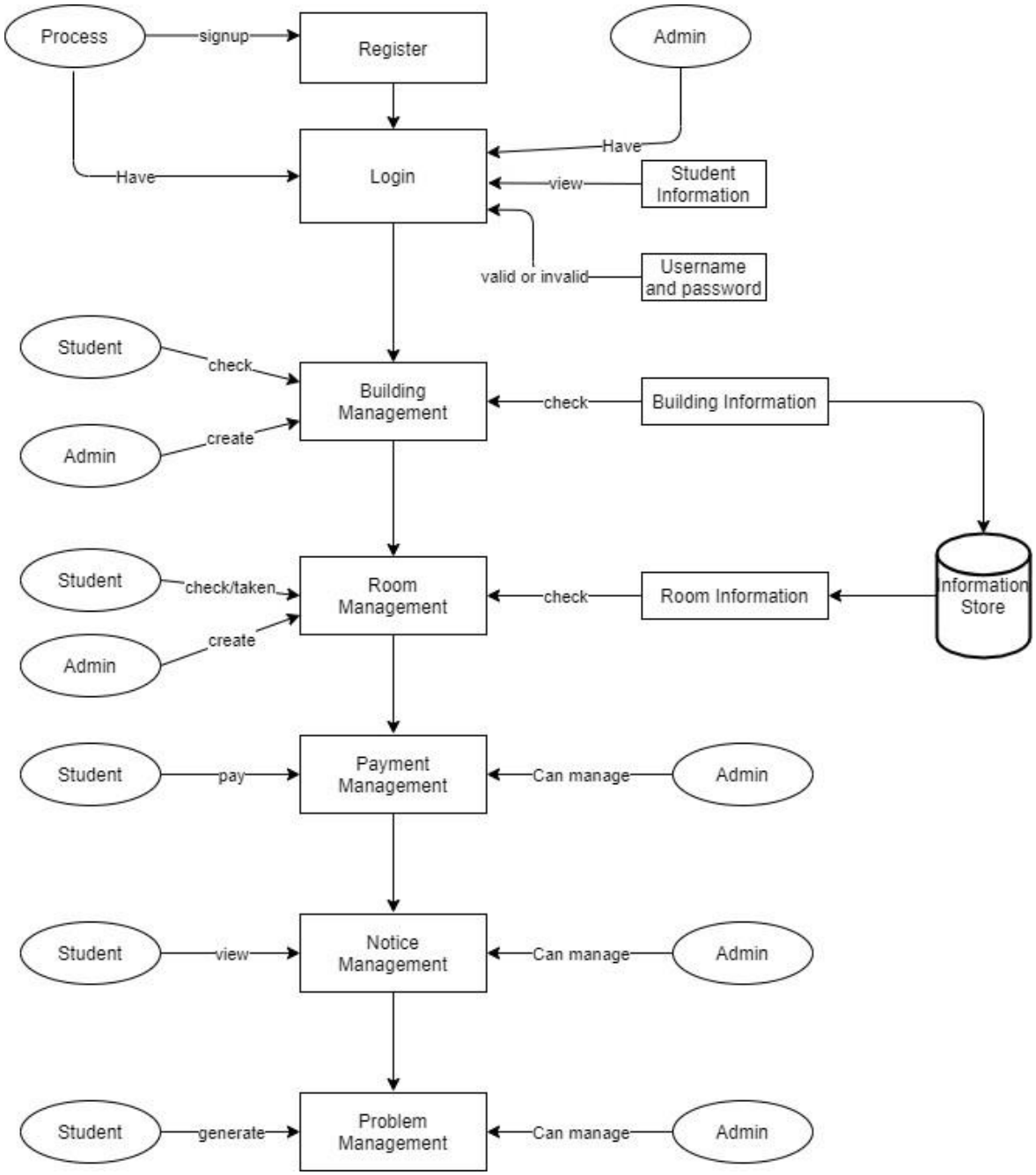



Figure 3.6.3 DFD 1 Level

# CHAPTER 4 SYSTEM DESIGN

## 4.1 Diagram of Activity

### 4.1.1 Student

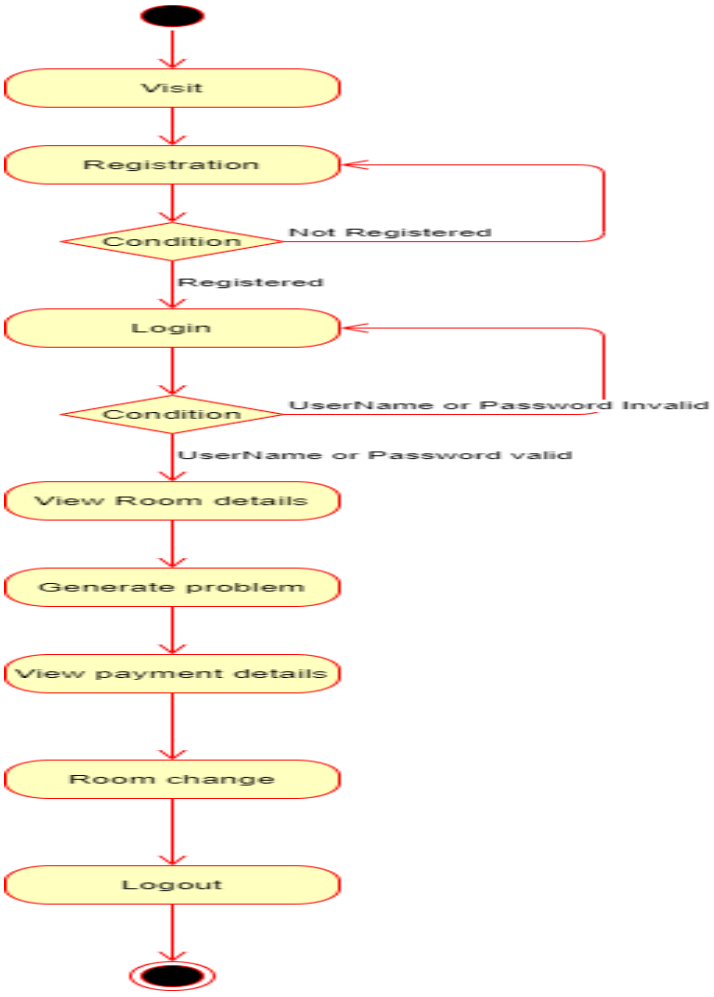



Figure 4.1.1 Student Activity Diagram

### 4.1.2 Admin

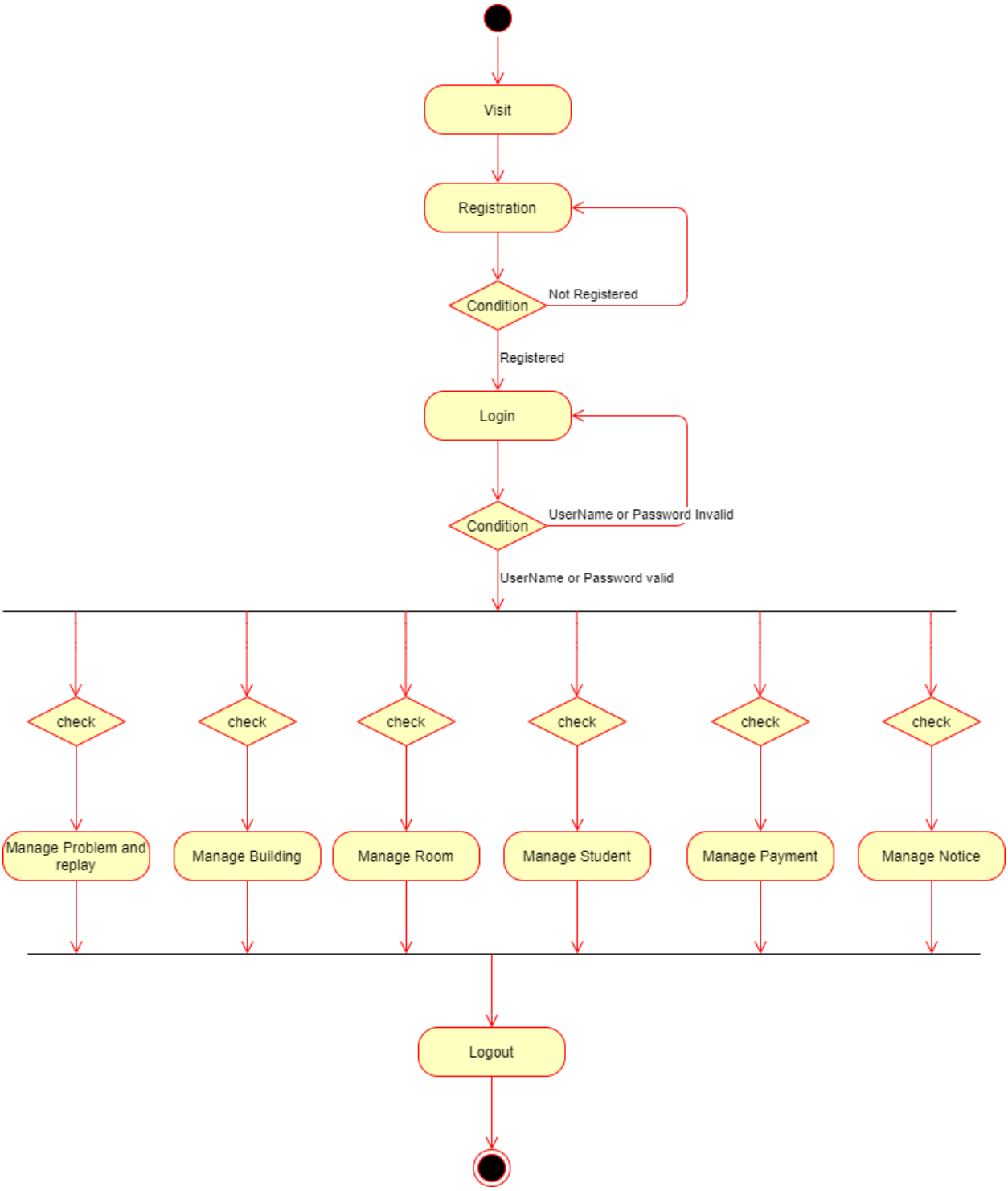



Figure 4.1.2 Admin Activity Diagram

## 4.2 Sequence Diagram

### 4.2.1 Login Operation of Admin

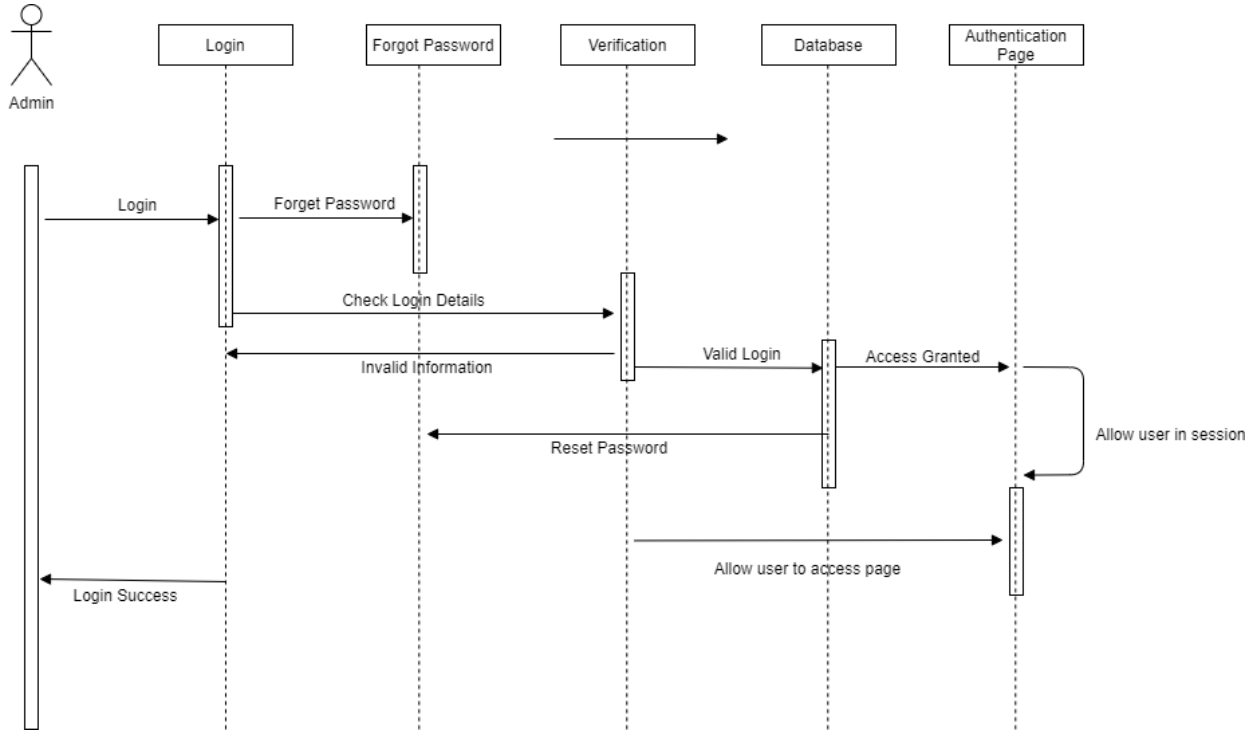



Figure 4.2.1 Login Operation of Admin

## 4.2.2 Admin Operation Process

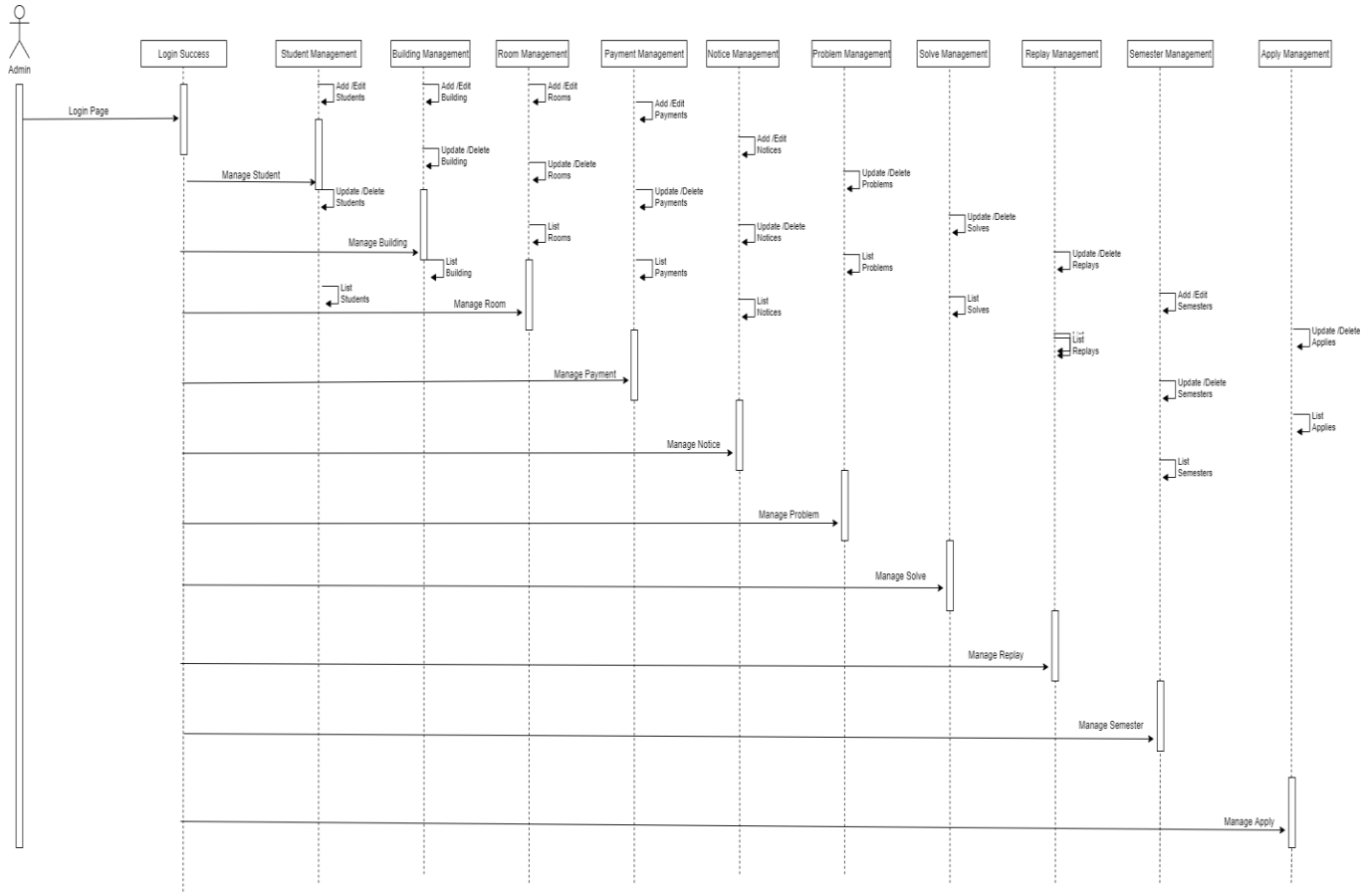



Figure 4.2.2 Admin Operation Process

### 4.2.3 Student Operation Process

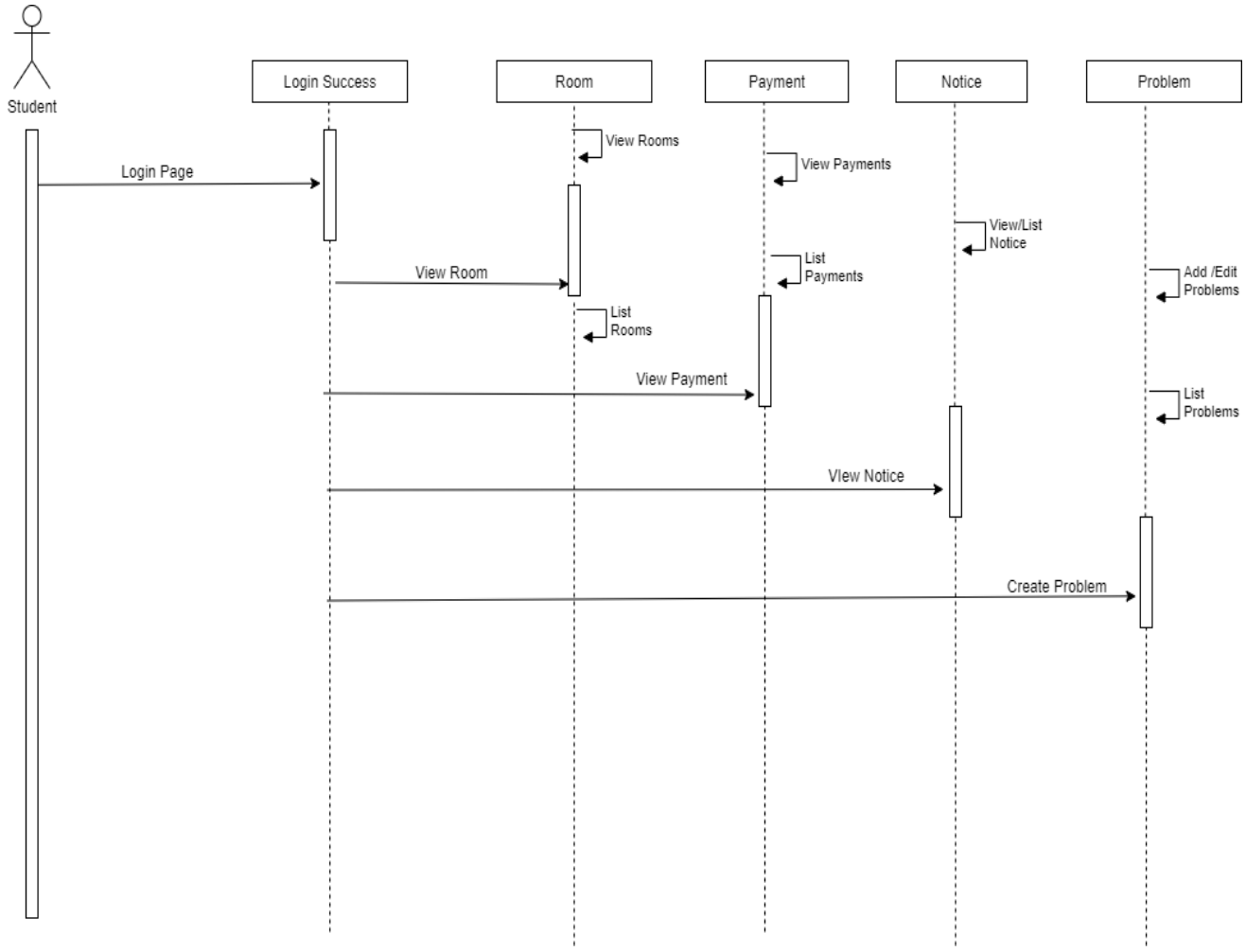



Figure 4.2.3 Student Operation Process

## 4.3 Front End Design

We use different type of markup language, style sheet, Programming language and some libraries forboth front-end and backend.

- HTML 5
- CSS
- Bootstrap4
- JavaScript

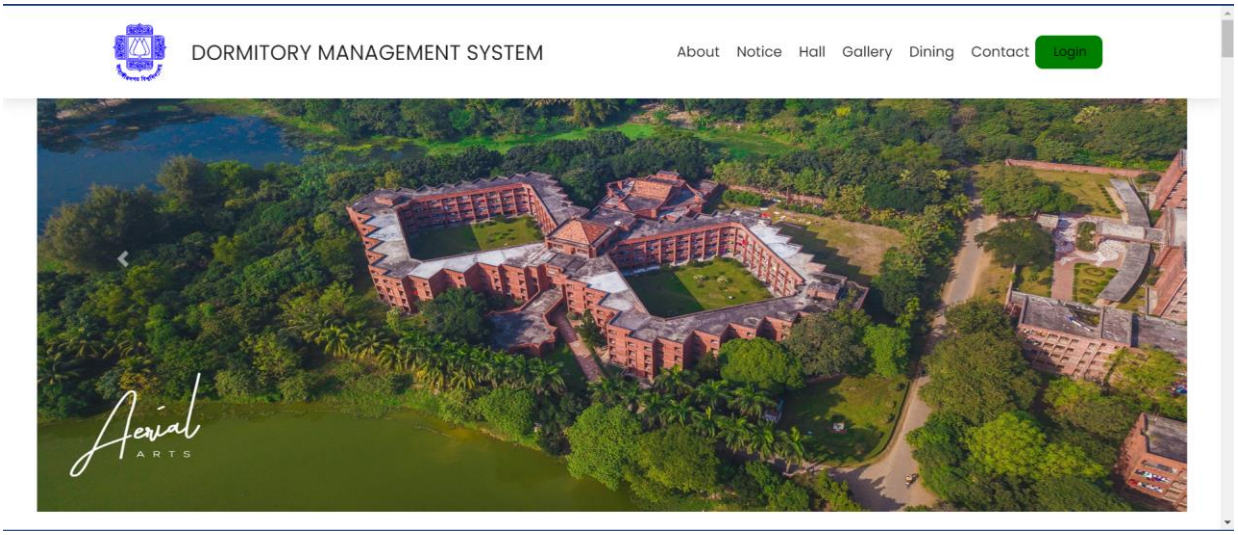



Figure 4.3.1 Home Page

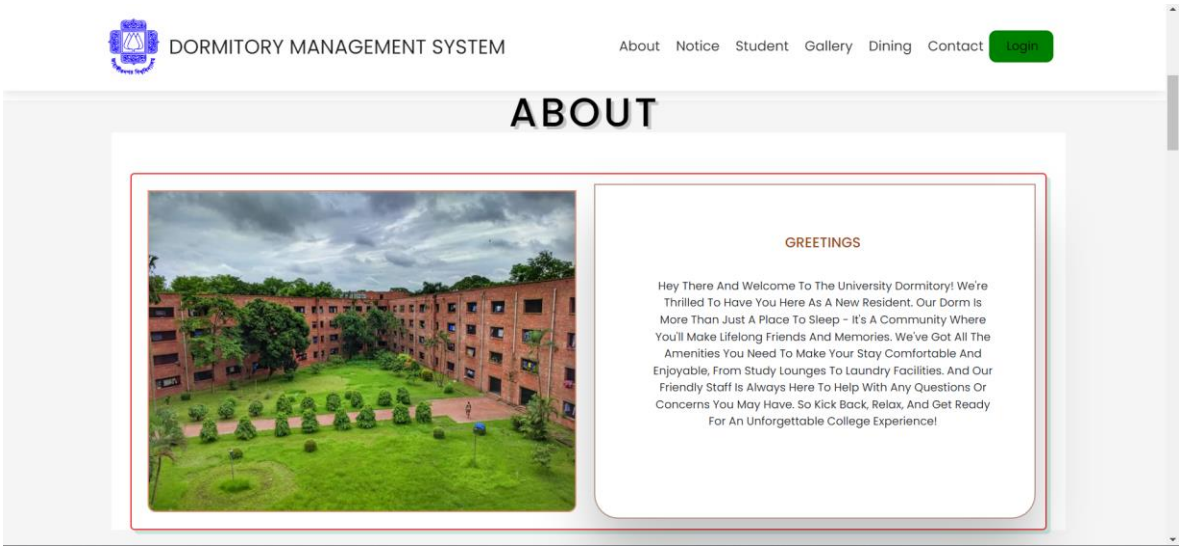



Figure 4.3.2 Our system overview


Figure 4.3.3 Hall list

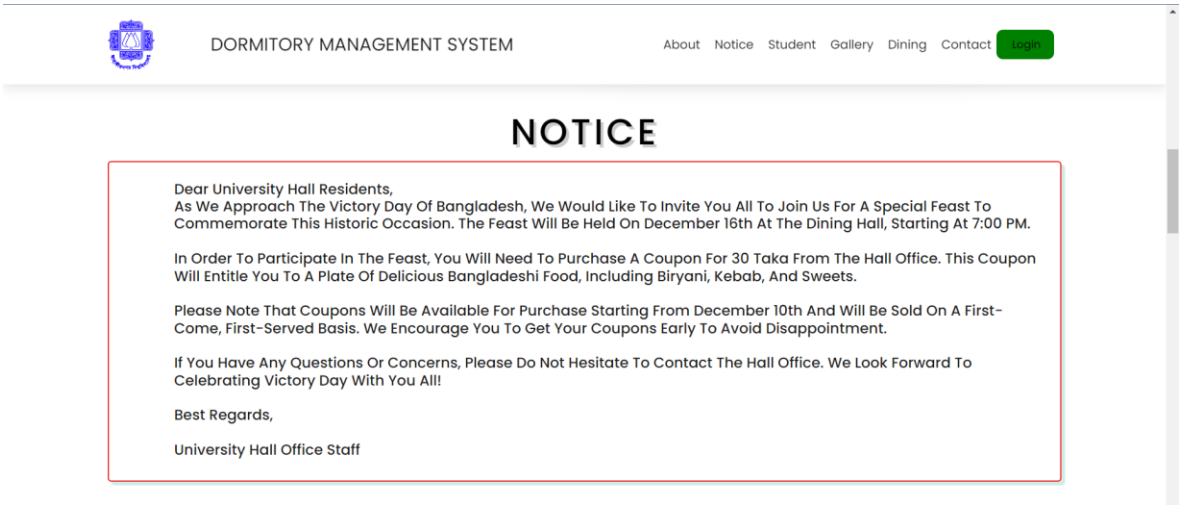



Figure 4.3.4 Notice

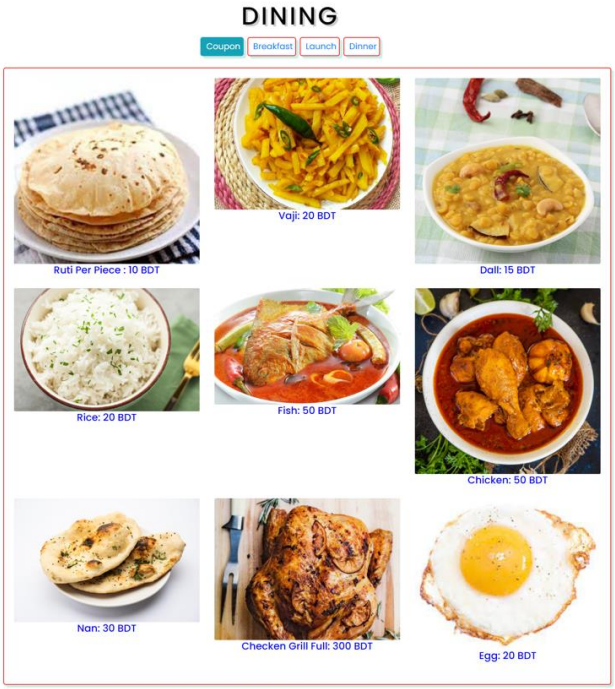



Figure 4.3.5 Dining

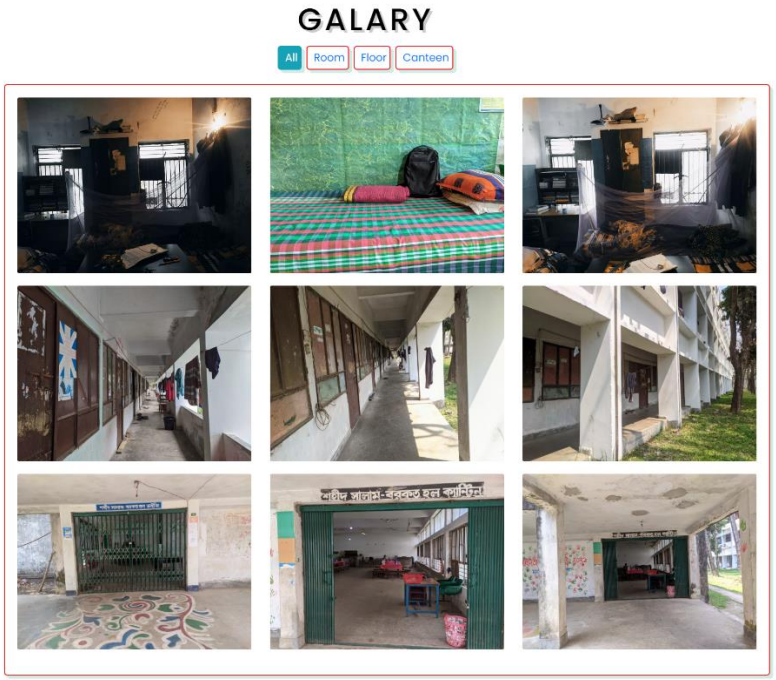



Figure 4.3.6 Gallery

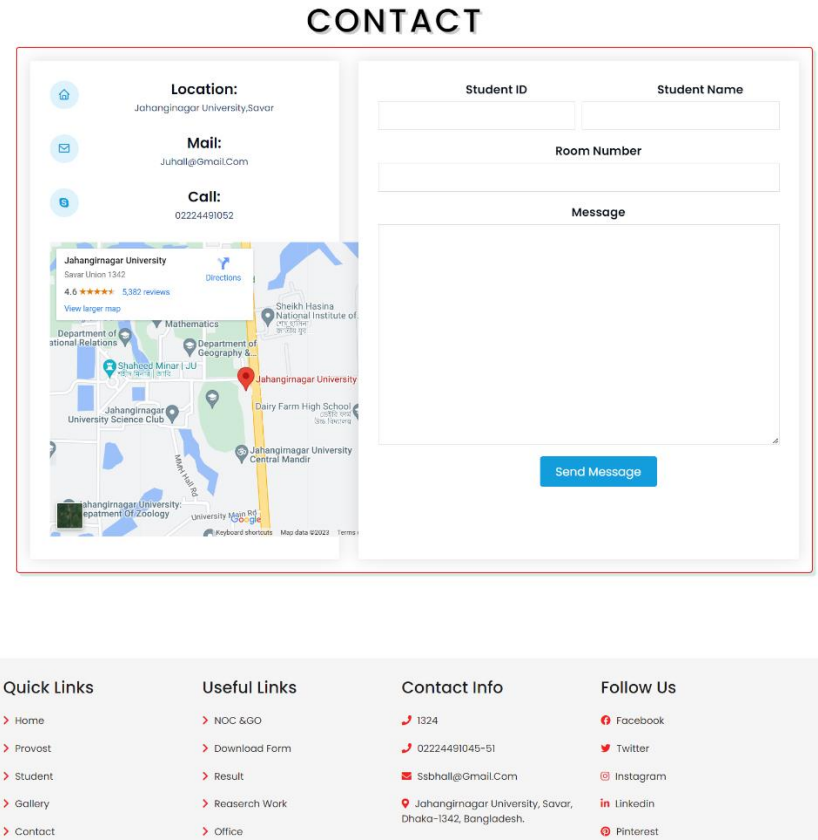



Figure 4.3.7 Contact

## 4.4 Back End Design

I used some programing languages and frameworks to complete the back-end design of thiswebsite.

- PHP
- Laravel Framework
- MySQL

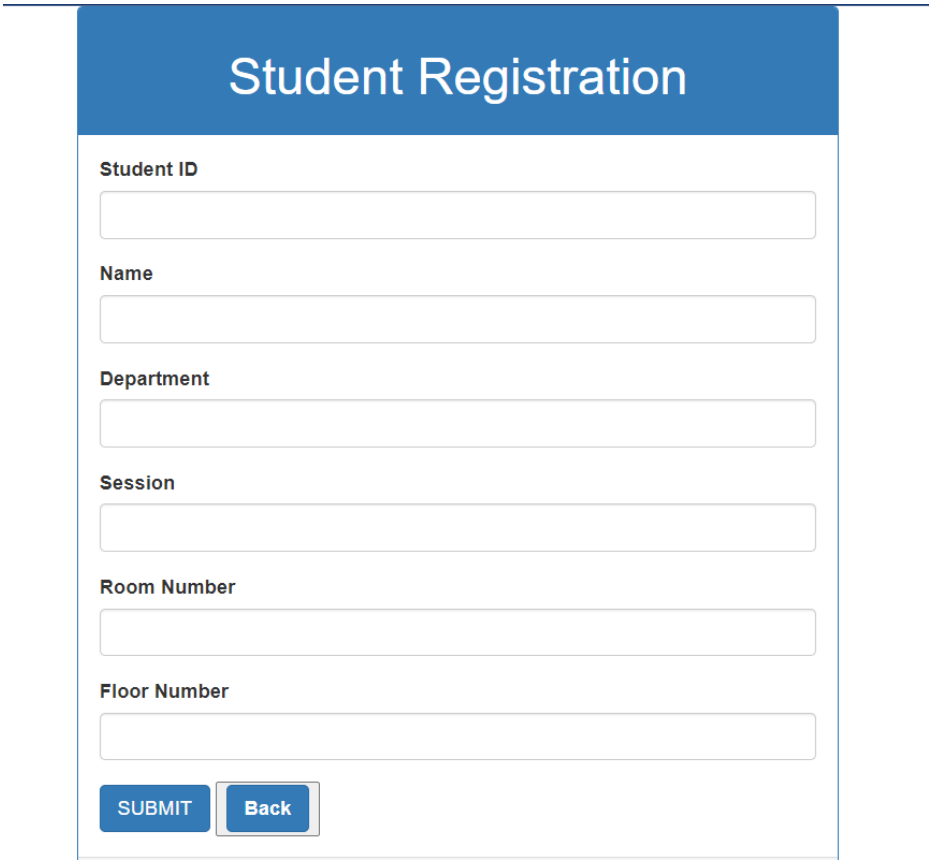



Figure 4.4.1 Student Registration Page

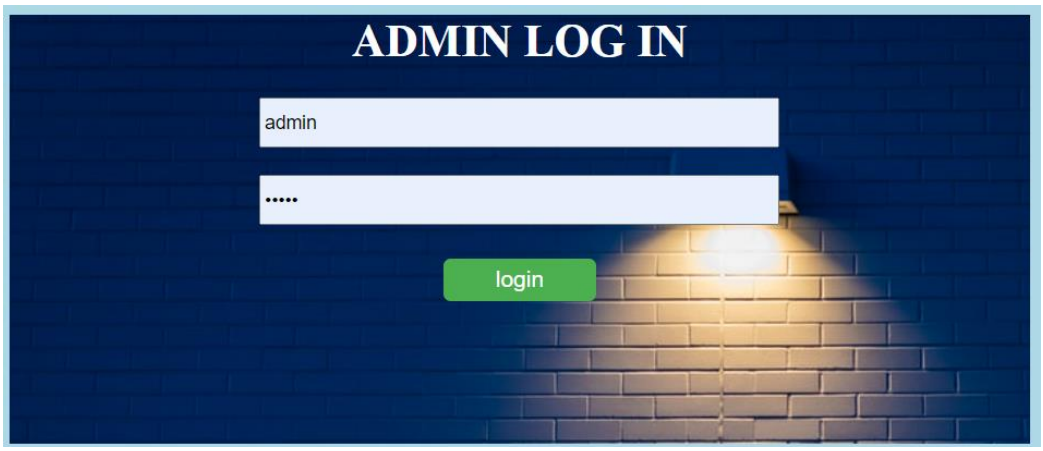



Figure 4.4.2 Admin Login Page.

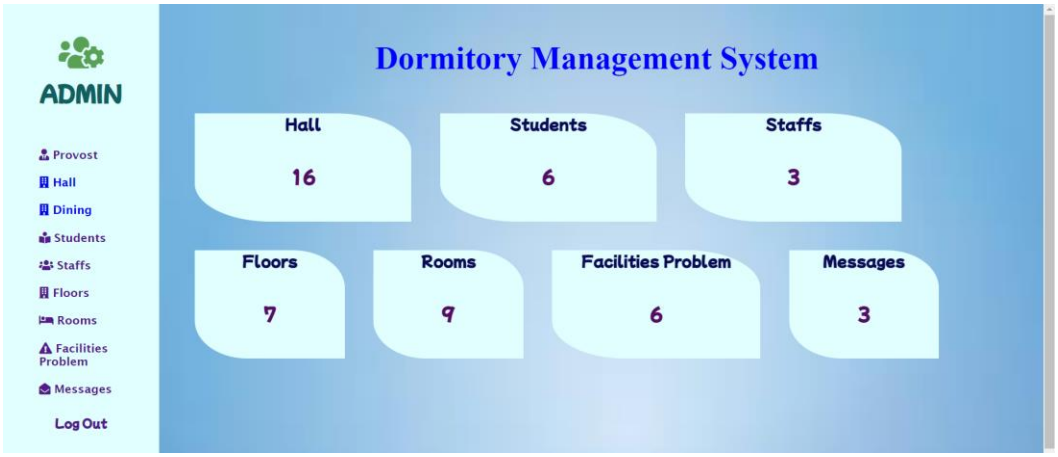



Figure 4.4.3 Admin Dashboard


Figure 4.4.4 Students Record

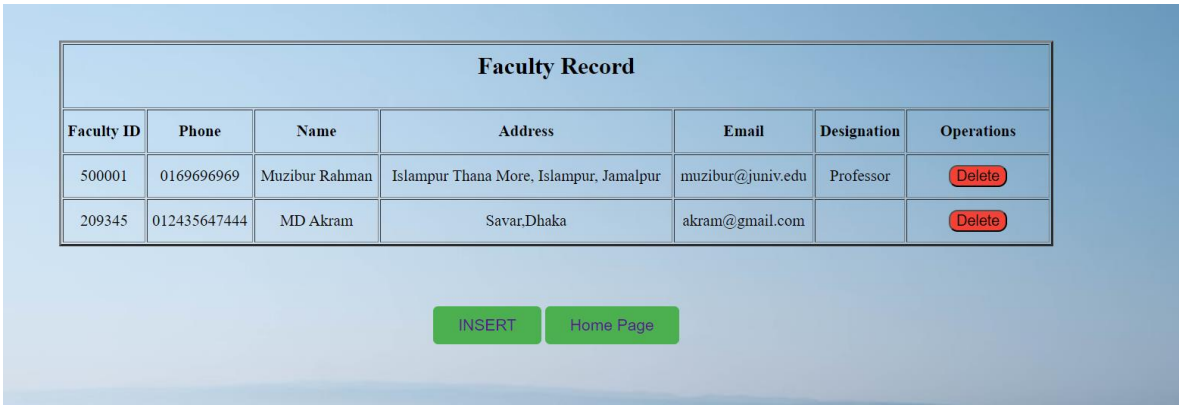



Figure 4.4.5 Faculty Member Information

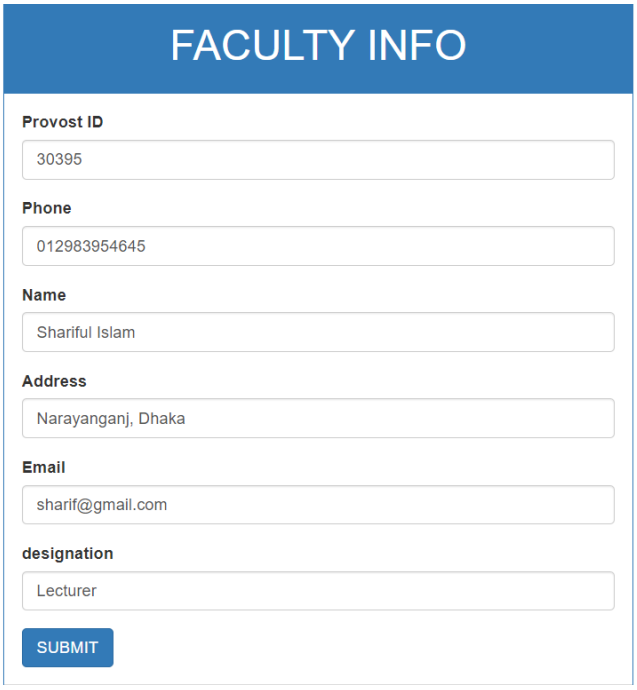



Figure 4.4.6 Add Faculty Member

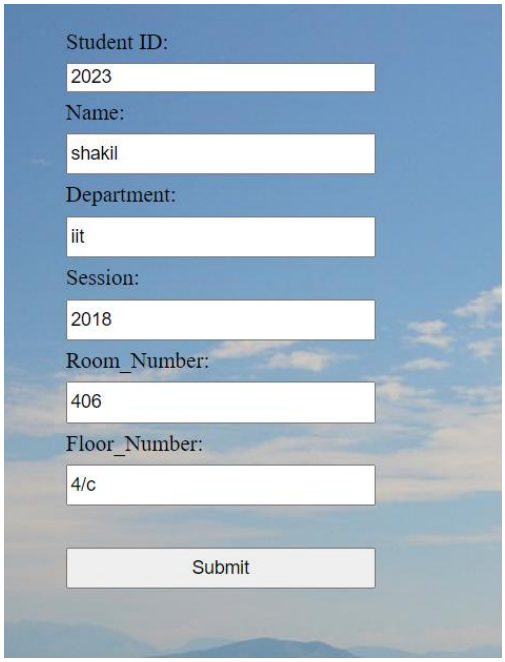



Figure 4.4.7 Update Student Information

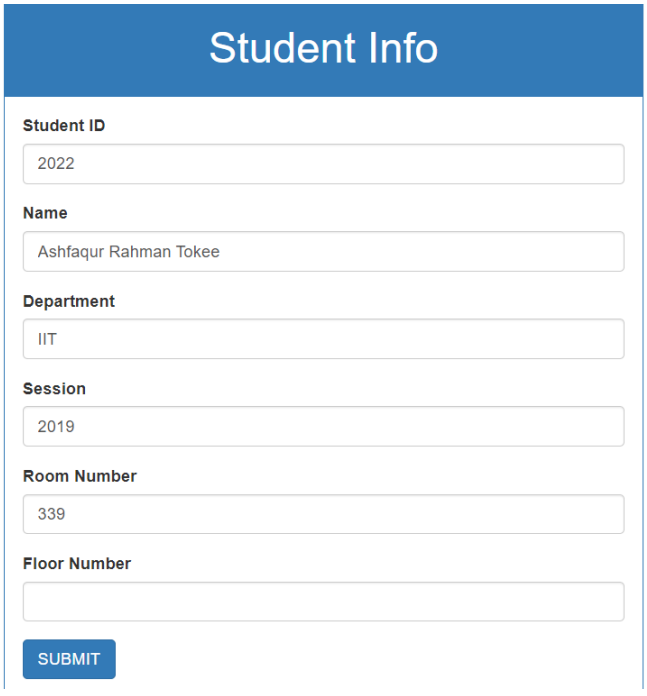



Figure 4.4.8 Add Student

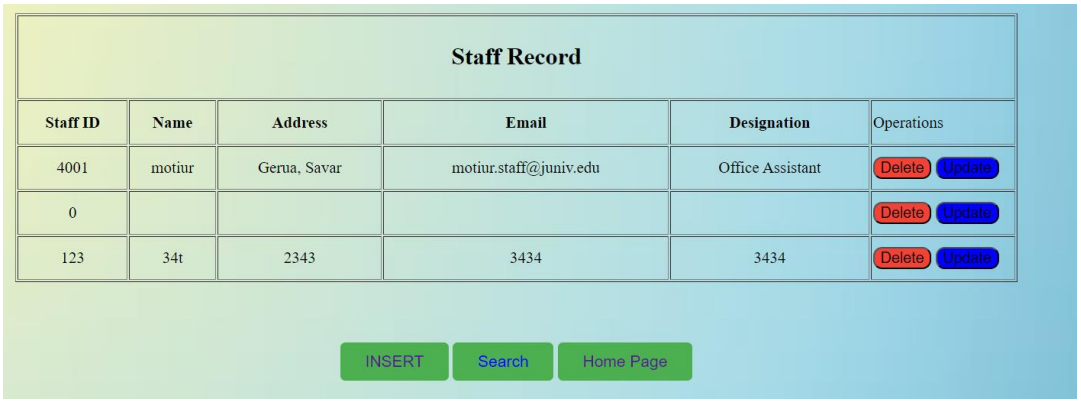



Figure 4.4.9 All Staffs

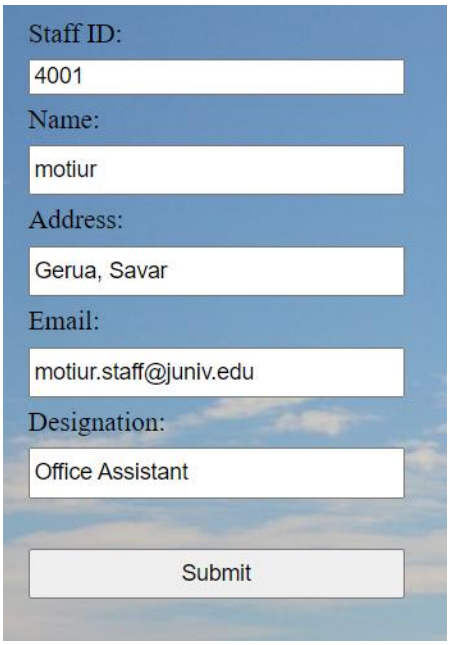



Figure 4.4.10 Update Staff Information

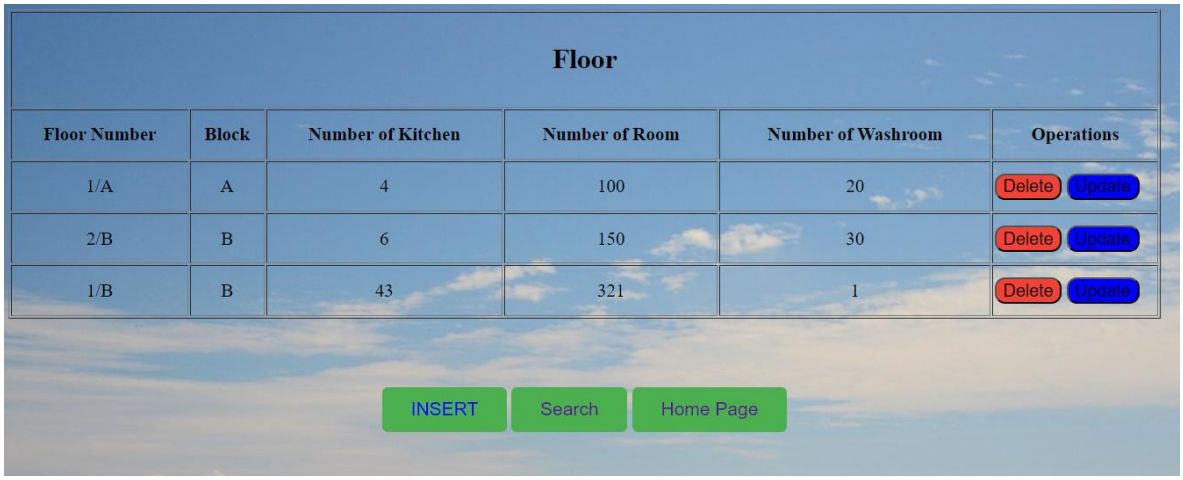



Figure 4.4.11 Floor information

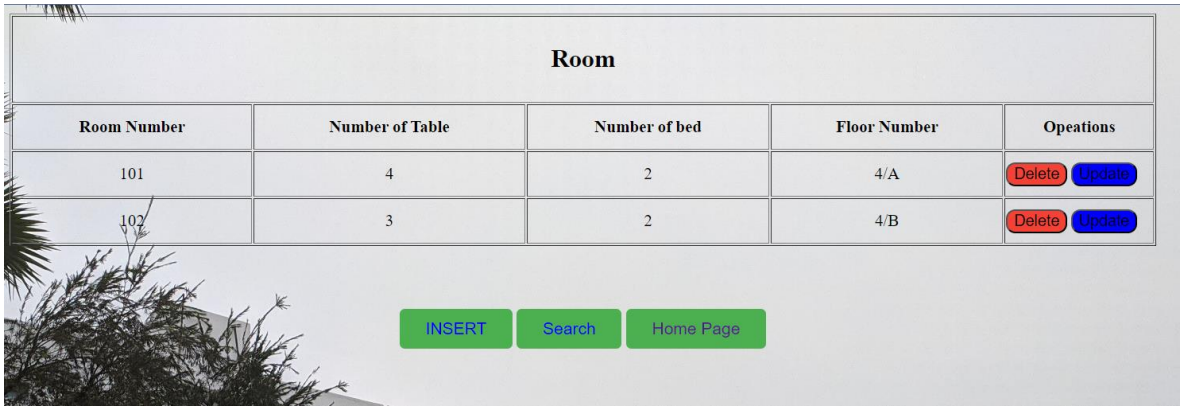



Figure 4.4.12 List of rooms

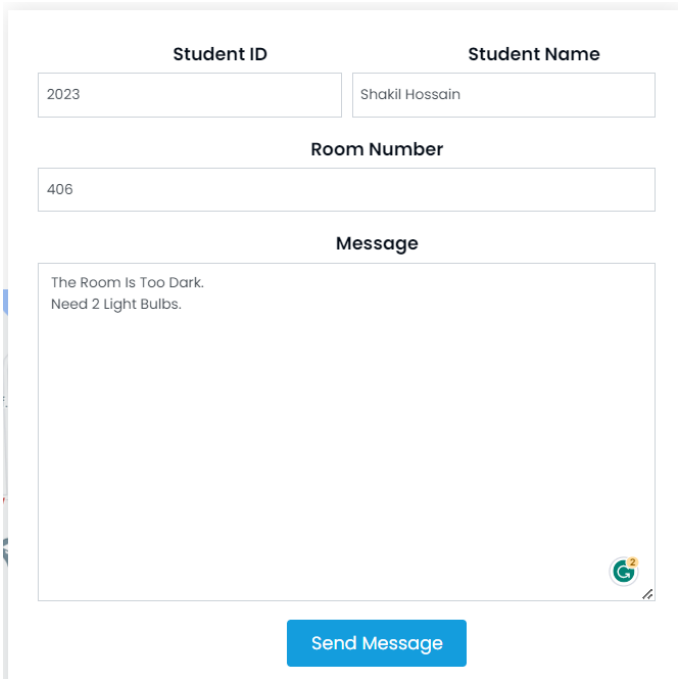
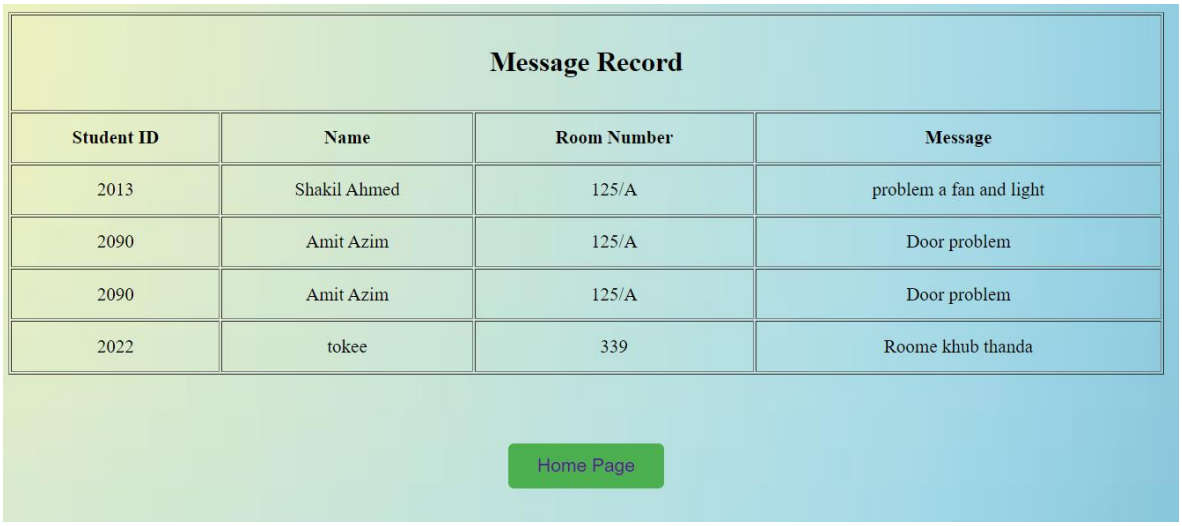



Figure 4.4.13 Issuing a complain



**Message Record**

| Student ID | Name | Room Number | Message |
|---|---|---|---|
| 2013 | Shakil Ahmed | 125/A | problem a fan and light |
| 2090 | Amit Azim | 125/A | Door problem |
| 2090 | Amit Azim | 125/A | Door problem |
| 2022 | tokee | 339 | Roome khub thanda |

Home Page

Figure 4.4.14 Complain List

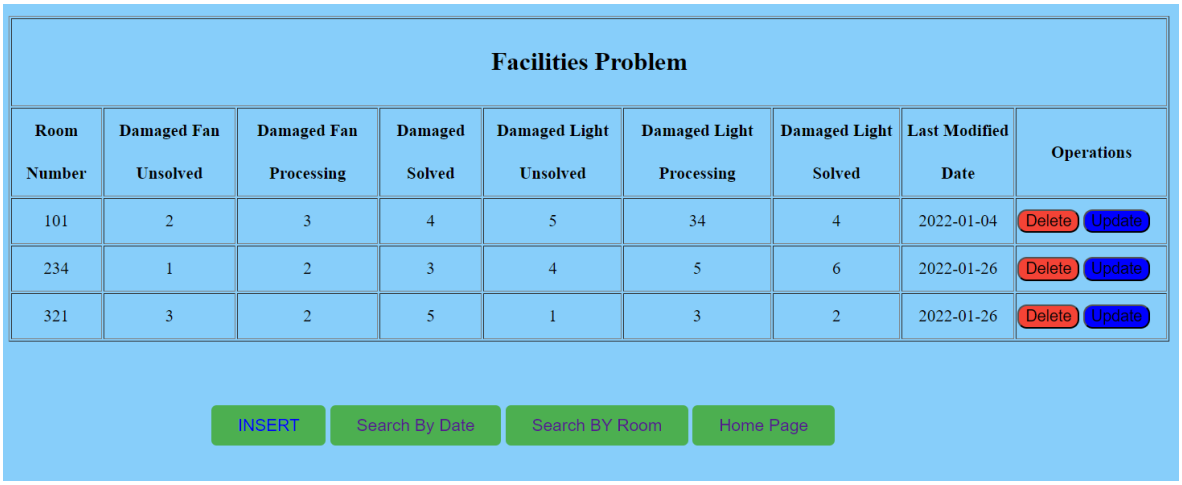



Figure 4.4.15 Evaluating Problems

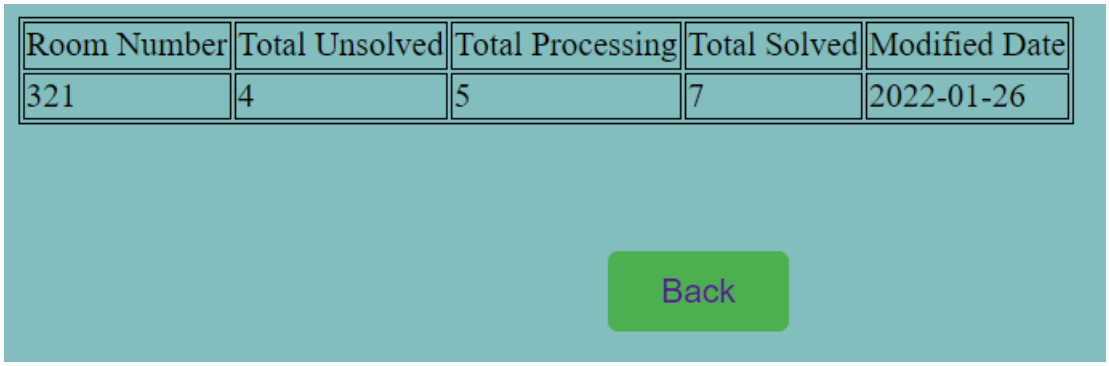



Figure 4.4.16 Search Problem by rooms

# CHAPTER 5

# IMPLEMENTATION AND TESTING

## 5.1 Database Implementation

TABLE 5.1.1 Database Details of Admin

| Chart Name | | Admin | | | | |
|---|---|---|---|---|---|---|
| Chart Statement | | This is chart container of Admin records | | | | |
| Area Name | Value Type | Volume | Not Null | PK | FK | Statement |
| Id | Int | 30 | √ | √ | | Container contain supreme id (Self Increase) |
| F_name | Varchar | 250 | √ | | | Container contain first name of admin |
| L_name | Varchar | 250 | | | | Container contain last name of admin |
| Email | Varchar | 250 | | | | Container contain email of admin |
| Contact | Varchar | 250 | | | | Container contain Contact of admin |
| Email_verified_at | Timestamp | | | | | Container contain email verification date of the admin |
| Password | Varchar | 250 | | | | Container contain password of admin |
| Is_active | Int | 1 | | | | Container contain active status of admin |
| Remember_token | varchar | 250 | | | | Container contain remember token of admin |

TABLE 5.1.2 Database Details of Applies

| Chart Name | | Applies | | | | |
|---|---|---|---|---|---|---|
| Chart Statement | | This is chart container of Applies records | | | | |
| Area Name | Value Type | Volume | Not Null | PK | FK | Statement |
| Id | Int | 30 | √ | √ | | Container contain supreme id (Self Increase) |
| St_application | Varchar | 250 | √ | | | Container contain student application |
| St_id | Int | 30 | | | √ | Container contain student id of apply |

TABLE 5.1.3 Database Details of Building

| Chart Name | | Building | | | | |
|---|---|---|---|---|---|---|
| Chart Statement | | This is chart container of Buildings records | | | | |
| Area Name | Value Type | Volume | Not Null | PK | FK | Statement |
| Id | Int | 30 | √ | √ | | Container contain supreme id (Self Increase) |
| Buiding_name | Varchar | 250 | √ | | | Container contain building name of building |

TABLE 5.1.4 Database Details of Members

| Chart Name | | Members | | | | |
|---|---|---|---|---|---|---|
| Chart Statement | | This is chart container of members record | | | | |
| Area Name | Value Type | Volume | Not Null | PK | FK | Statement |
| Id | Int | 30 | √ | √ | | Container contain supreme id (Self Increment) |
| Name | Varchar | 250 | √ | | | Container contain name of member |
| Email | Varchar | 250 | √ | | | Container contain email of member |
| Position | Varchar | 250 | √ | | | Container contain position of member |
| Contact | Int | 30 | √ | | | Container contain contact of member |
| Image | Varchar | 250 | √ | | | Container contain image of member |

TABLE 5.1.5 Database Details of Notice

| Chart Name | | Notice | | | | |
|---|---|---|---|---|---|---|
| Chart Statement | | This is chart container of notice records | | | | |
| Area Name | Value Type | Volume | Not Null | PK | FK | Statement |
| Id | Int | 30 | √ | √ | | Container contain id (Self Increase) |
| Title | Varchar | 250 | √ | | | Container contain notice title of notice |
| File | Varchar | 250 | √ | | | Container contain file of notice |

TABLE 5.1.6 Database Information Details of Reset

| Chart Name | | Password Reset | | | | |
|---|---|---|---|---|---|---|
| Chart Statement | | This is chart container of password reset records | | | | |
| Area Name | Value Type | Volume | Not Null | PK | FK | Statement |
| Email | VARCHAR | 250 | √ | | | Container contain email of password reset |
| Token | VARCHAR | 250 | √ | | | Container contain token of password reset |

TABLE 5.1.7 Database Details of Payments

| Chart Name | | Payments | | | | |
|---|---|---|---|---|---|---|
| Chart Statement | | This is chart container of payments records | | | | |
| Area Name | Value Type | Volume | Not Null | PK | FK | Statement |
| Id | Int | 30 | √ | √ | | Container contain supreme id (Self Increase) |
| St_id | Int | 30 | √ | | √ | Container contain student id of payments |
| St_semester | Varchar | 250 | √ | | | Container contain student semester of payments |
| Hall_fee | Int | 30 | √ | | | Container contain hall fee of payments |

TABLE 5.1.8 Database Details of Problems

| Chart Name | | Problems | | | | |
|---|---|---|---|---|---|---|
| Chart Statement | | This is chart container of problems record | | | | |
| Area Name | Value Type | Volume | Not Null | PK | FK | Statement |
| Id | Int | 30 | √ | √ | | Container contain supreme id (Self Increase) |
| P_description | Varchar | 250 | √ | | | Container contain problem description of problems |
| St_id | Int | 30 | √ | | √ | Container contain student id of problems |

TABLE 5.1.9 Database Details of Replies

| Chart Name | | Replies | | | | |
|---|---|---|---|---|---|---|
| Chart Statement | | This is chart container of replies records | | | | |
| Area Name | Value Type | Volume | Not Null | PK | FK | Statement |
| Id | Int | 30 | √ | √ | | Container contain supreme id (Self Increase) |
| St_reply | Varchar | 250 | √ | | | Container contain student reply of replies |
| Problem_id | Int | 30 | | | | Container contain problem id of replies |

TABLE 5.1.10 Database Details of Rooms

| Chart Name | | Rooms | | | | |
|---|---|---|---|---|---|---|
| Chart Statement | | This is chart container of rooms records | | | | |
| Area Name | Value Type | Volume | Not Null | PK | FK | Statement |
| Id | Int | 30 | √ | √ | | Container contain supreme id (Self Increase) |
| Room_num | Varchar | 250 | √ | | | Container contain room name of rooms |
| Quantity | Int | 30 | √ | | | Container contain quantity of rooms |
| Booked | Int | 30 | √ | | | Container contain booked status of rooms |
| Building_id | Int | 30 | √ | | √ | Container contain building id of rooms |

TABLE 5.1.11 Database Details of Semester

| Chart Name | | Semester | | | | |
|---|---|---|---|---|---|---|
| Chart Statement | | This is chart container of semester records | | | | |
| Area Name | Value Type | Volume | Not Null | PK | FK | Statement |
| Id | Int | 30 | √ | √ | | Container contain supreme id (Self Increase) |
| Semester_name | Varchar | 250 | √ | | | Container contain semester name of semester |

TABLE 5.1.12 Database Details of Slider

| Chart Name | | Slider | | | | |
|---|---|---|---|---|---|---|
| Chart Statement | | This is chart container of slider records | | | | |
| Area Name | Value Type | Volume | Not Null | PK | FK | Statement |
| Id | BIG (INT) | 30 | √ | √ | | Container contain supreme id (Self Increase) |
| Image | VARCHAR | 250 | √ | | | Container contain image of slider |

TABLE 5.1.13 Database Details of Student

| Chart Name | | Student | | | | |
|---|---|---|---|---|---|---|
| Chart Statement | | This is chart container of student records | | | | |
| Area Name | Value Type | Volume | Not Null | PK | FK | Statement |
| St_id | Int | 250 | √ | √ | | Container supreme id (Self Increase) |
| St_name | Varchar | 250 | √ | | | Container contain student name of student |
| Email | Varchar | 250 | √ | | | Container contain email of student |
| Password | Varchar | 250 | √ | | | Container contain password of student |
| St_dept | Varchar | 250 | √ | | | Container contain student department of student |
| Image | Varchar | 250 | √ | | | Container contain mage of student |
| Room_id | Int | 30 | √ | | √ | Container contain room id of student |
| Semester_id | Int | 30 | √ | | √ | Container contain semester id of student |

| | | | | | | |
|---|---|---|---|---|---|---|
| St_contact | Varchar | 250 | √ | | | Container contain student contact of student |
| Remember_token | Varchar | 250 | √ | | | Container contain remember token of student |

TABLE 5.1.14 Database Details of User

| Chart Name | | User | | | | |
|---|---|---|---|---|---|---|
| Chart Statement | | This is chart container of user records | | | | |
| Area Name | Value type | Volume | Not Null | PK | FK | Statement |
| Id | int | 30 | √ | √ | | Container supreme id (Self Increase) |
| F_name | varchar | 250 | √ | | | Container contain fast name of user |
| L_name | varchar | 250 | √ | | | Container contain last name of user |
| Email | varchar | 250 | √ | | | Container contain email of user |
| Contact | varchar | 250 | √ | | | Container contain contact of user |
| Address | varchar | 250 | √ | | | Container contain address of user |
| Email_verified_at | Timestamp | | | | | Container contain email verified time of user |
| Password | varchar | 250 | √ | | | Container contain password of user |
| Remember_token | varchar | 250 | √ | | | Container contain remember token of user |

## 5.2 Test Case

TABLE 5.2.1 Login Page of Test Case Description

| Serial No | Input/Action | Desired Value | Indeed Value | Comment |
|---|---|---|---|---|
| 1 | Permit the field empty | The field email and password required | Message “Email and password is required” | Granted |
| 2 | Taken ineffective Password | Password is incorrect | Message “Password is incorrect” | Granted |
| 3 | Taken ineffective email format | Please enter a valid email | Message “Please enter a valid email” | Granted |
| 4 | Taken acceptable user name or email | Accepted Value | Value accepted | Granted |

TABLE 5.2.2 Registration Page of Test Case Description

| Serial No | Activity | Desired Value | Indeed Value | Comment |
|---|---|---|---|---|
| 1 | Permit any field empty | This field is empty | Message “This field is empty” | Granted |
| 2 | Taken an ineffective email or already used | This email is invalid or try with another | Message “Enter a valid email or try with another” | Granted |
| 3 | Taken an ineffective phone number | Phone number is invalid | Message “Phone number is invalid” | Granted |
| 4 | Taken valid data | Accepted Value | Value accepted | Granted |

# CHAPTER 6

## CONCLUSON AND FUTURE WORK

### 6.1 Future Works

We want our system to be used in our university campus. Then in future if it is convenient to use by the users, we will try to make it as a versatile system. And the problems we have faced, in future we have planned to solve this problem. We have a well-planned idea about it. Some possible future works would include.

- Integration with other campus systems: In the future, you could explore integrating the dormitory management system with other campus systems, such as the student information system or the campus security system, to create a more comprehensive and interconnected platform for managing student life on campus.
- Introduce a payment gateway system like Bikash, Nogod, DBBL.
- Automated room assignment algorithms: To improve the efficiency and accuracy of room assignments, you could develop more advanced algorithms that take into account factors such as student preferences, availability, and compatibility.
- Mobile app development: To make the system more accessible and convenient for users, you could develop a mobile app version of the website that allows students and administrators to access the system on-the-go.
- Smart building integration: In the future, you could explore integrating the dormitory management system with smart building technology, such as sensors and automation systems, to create a more energy-efficient and environmentally-friendly campus.
- Social media integration: To improve communication and engagement with students, you could explore integrating the dormitory management system with social media platforms, such as Facebook and Twitter, to allow for more seamless communication and outreach.

## 6.2 Conclusion

Dormitory management systems are software tools that streamline the management of dormitories and student housing facilities. These systems typically include features such as room assignment and scheduling, facility maintenance and repairs, rent payment tracking, and communication tools for residents and staff.

Overall, dormitory management systems can greatly benefit both students and staff by increasing efficiency and reducing administrative workload. By automating many of the tasks associated with managing a dormitory, staff members can devote more time to providing support and resources to students.

Additionally, these systems often provide valuable data and analytics that can inform decision-making around facility management and student programming. For example, usage patterns and feedback from residents can help staff optimize resource allocation and improve the overall living experience for students.

In summary, dormitory management systems offer a variety of benefits for both students and staff, including improved efficiency, better communication, and data-driven decision-making. As such, they are increasingly becoming a standard tool for managing student housing facilities.

# REFERENCES


[1] “Dhaka University Hall Management Web Portal” du.ac.bd, 2023. [Online]. Available: http://www.du.ac.bd/home/hall_admin/hall. [Accessed: Jan. 03, 2023]

[2] “Shahjalal University of Science and Technology Hall” sust.edu, 2023. [Online]. Available: http://www.sust.edu/campus-life/residence-hall. [Accessed: Jan. 06, 2023]

[3] “Khulna University of Engineering and Technology” kuet.ac.bd, 2023. [Online]. Available: http://www.kuet.ac.bd/index.php/welcome/khajahalldetails. [Accessed: Feb. 03, 2023]

[4] "Design and Implementation of a Dormitory Management System Based on Web Services." by Hongbin Wang, Journal of Physics: Conference Series, Volume 1156, Number 4 (2019). [Accessed: Apr. 03, 2023]

[5] "Development of a Web-Based Student Housing Management System." by Gopal Krishna Shrestha and Deepak Tamang, International Journal of Computer Science and Network Security, Volume 18, Number 5 (2018). [Accessed: Mar. 02, 2023]

[6] "Design and Implementation of a Smart Dormitory Management System Based on the Internet of Things." by Yanhui Zhu, International Journal of Future Computer and Communication, Volume 8, Number 1 (2019). [Accessed: Apr. 01, 2023]

[7] "Design and Development of Hostel Management System using Web Technology." by R. Rajeswari and K. R. Sridevi, International Journal of Computer Science and Information Technologies, Volume 6, Number 3 (2015). [Accessed: Feb. 03, 2023]

[8] "Design and Development of an Automated Dormitory Management System." by A. S. Adegboye and A. T. Adebowale, International Journal of Computer Science and Mobile Computing, Volume 8, Number 7 (2019). [Accessed: Apr. 03, 2023]

[9] "A Web-Based Dormitory Management System for Higher Education Institutions." by C. L. Bautista and M. F. L. Asistio, International Journal of Computer Applications, Volume 179, Number 45 (2018). [Accessed: Maf. 30, 2023]

[10] "Development of a Web-Based Hostel Management System for Tertiary Institutions." by G. O. Okegbemi and S. O. Oyebisi, Journal of Engineering and Applied Sciences, Volume 12, Number 1 (2017). [Accessed: Apr. 04, 2023]